



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/657,989	09/09/2003	Shree A. Dandekar	DC-05164	1287
33438	7590	03/27/2008		
HAMILTON & TERRILE, LLP P.O. BOX 203518 AUSTIN, TX 78720			EXAMINER CHEN, QING	
			ART UNIT 2191	PAPER NUMBER
			NOTIFICATION DATE 03/27/2008	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docketing@hamiltontertile.com
seaton@hamiltontertile.com
tmunoz@hamiltontertile.com



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/657,989
Filing Date: September 09, 2003
Appellant(s): DANDEKAR ET AL.

Gary W. Hamilton (Reg. No. 31,834)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed on January 10, 2008 appealing from the Office action mailed on April 9, 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The Examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The Appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The Appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,991,543	AMBERG et al.	11-1999
6,075,943	FEINMAN et al.	6-2000
6,088,803	TSO et al.	7-2000
6,378,054	KARASUDANI et al.	4-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Appellant has requested that the provisional obviousness-type double patenting rejections of Claims 1, 3, 5-8, 10, 12-15, 17, 19, and 20 be held in abeyance until receipt of an indication of allowable subject matter. Therefore, the aforementioned rejection is not submitted for review in this appeal.

1. **Claims 1, 3, 5, 6, 8, 10, 12, 13, 15, 17, 19, and 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Amberg et al.** (US 5,991,543) in view of **Feinman** (US 6,075,943).

As per **Claim 1**, Amberg et al. disclose:

- a distribution server operable to receive a software file (*see Figure 1: 140; Column 3: 48-55, "... sequencing program 204 residing on step maker 140, first reads a plurality of component descriptors from descriptor file 96."* and 59-66, "... the component descriptors are included in a descriptor file called a system descriptor record which is a computer readable file containing a listing of the components, hardware and/or software components, to be installed onto target system 160.");

- a repack and script regeneration server operably connected to said distribution server, said repack and script regeneration server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software file (*see Figure 1: 140; Column 4: 8-14, "Having retrieved the software installation and/or testing steps appropriate for target system 160, sequencing program 204 sequences the steps in a predetermined order according to sequence numbers corresponding to each step. Having sequenced the steps required for target system 160, sequencing program 204 writes a series of output files to step disk 150."*; Column 9: 56-67 through Column 10: 1-2, "The result of the joinder of the Sys_Comp table 112 and the Comp_Step table 114 is then joined with the Sys_Step_Seq table 106 which contains all the steps for family X." and "... a three-table join of Sys_Comp table 112, Comp_Step table 114, and Sys_Step_Seq table 106 yields the appropriate software installation and testing steps as well as sequencing information in the form of sequence and phase numbers to install and/or test software upon target computer system 160.");

- a compliance server operably connected to said distribution server, said compliance server being operable to perform compliance verification to confirm that said software file complies with a predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, "... the*

Art Unit: 2100

sequencing program and database determine if the processor, hard drive, monitor, and software contained in the system descriptor record of FIG. 3B have corresponding entries and corresponding integers specified by CompID in Component table 108.”); and

- a download server operable to transfer said software file to a target information handling system after verifying that said software file complies with a predetermined set of rules (see Figure 1: 170 and 190; Figure 2: 202; Column 10: 61-64, “... the output files reside upon the server 202 or file server 190, where they can be used to direct the execution of the software installation and testing steps upon target computer system 160.”).

However, Amberg et al. do not disclose:

- a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system.

Feinman discloses:

- a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system (see Figure 1B: 11; Figure 7: 100; Column 3: 3-5, “... the server 11 shown has application programs which require automatic delivery to the remote clients 13.” and 44-67 through Column 4: 1-4, “Once the application program(s) have been packed up, the next step is to identify the remote client’s delivery points, the application programs that are to be

Art Unit: 2100

delivered to each delivery point and the time that each application program is to be delivered (FIG. 3).”).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to a target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to send the application program to the remote site at a specified time, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Feinman – Column 1: 23-32*).

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and Amberg et al. further disclose:

- wherein said compliance server is operable to automatically generate a non-compliance notice message upon detection that said software file does not comply with said predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, “If a component is not legal (i.e. if a component in the system descriptor record is not contained in Component table 108), an error flag is raised.”*).

As per **Claim 5**, the rejection of **Claim 1** is incorporated; and Amberg et al. further disclose:

- a test control server operable to verify proper operation of said software file on said target information handling system (*see Figure 2: 202; Figure 11; Column 12: 1-3, "... this program is called Runstep and is located on step disk 150 in the embodiment of FIG. 1 and on file server 202 in the embodiment of FIG. 2." and 62-67 through Column 13: 1-12, "... the Runstep program fist checks to see if a file named Re_Run.bat exists. A Re_Run.bat file is created before any command is executed from a step sequence and is removed after successful completion of the command. The existence of Re_Run.bat indicates to the Runstep program in module 900 that the last command run was not successfully completed."*).

However, Amberg et al. do not disclose:

- a test control server operable to confirm the download of said software file to said target information handling system.

Feinman discloses:

- a test control server operable to confirm the download of said software file to said target information handling system (*see Figure 5: 70; Column 3: 40-43, "The list will be used later on by the automatic installation system to ensure that all files for an installation were transferred and unpacked successfully."; Column 5: 18-21, "If the file transfer fails in step 70, an error code is trapped and an error message is placed into two of three logs maintained by the automatic installation system as shown in step 74."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to

Art Unit: 2100

include a test control server operable to confirm the download of said software file to said target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to validate that the application delivery was successful, so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Feinman – Column 1: 33-37*).

As per **Claim 6**, the rejection of **Claim 1** is incorporated; and Amberg et al. further disclose:

- wherein said distribution server is operable to notify a manager regarding the status of said software file within a software distribution system (*see Column 14: 20-25, “... results from the software installation and testing may be logged back to either file server 190 or to file server 202. The results preferably include whether all the steps were completed successfully and what types of failures (if any) were encountered.”*).

As per **Claim 8**, Amberg et al. disclose:

- receiving a software file (*see Figure 1: 140; Column 3: 48-55, “... sequencing program 204 residing on step maker 140, first reads a plurality of component descriptors from descriptor file 96.” and 59-66, “... the component descriptors are included in a descriptor file called a system descriptor record which is a computer readable file containing a listing of the components, hardware and/or software components, to be installed onto target system 160.”*);
- disassembling said software file and repackaging said software file with scripts for automatically controlling the transfer of said software file (*see Figure 1: 140; Column 4: 8-14,*

Art Unit: 2100

“Having retrieved the software installation and/or testing steps appropriate for target system 160, sequencing program 204 sequences the steps in a predetermined order according to sequence numbers corresponding to each step. Having sequenced the steps required for target system 160, sequencing program 204 writes a series of output files to step disk 150.”; Column 9: 56-67 through Column 10: 1-2, “The result of the joinder of the Sys_Comp table 112 and the Comp_Step table 114 is then joined with the Sys_Step_Seq table 106 which contains all the steps for family X.” and “... a three-table join of Sys_Comp table 112, Comp_Step table 114, and Sys_Step_Seq table 106 yields the appropriate software installation and testing steps as well as sequencing information in the form of sequence and phase numbers to install and/or test software upon target computer system 160.”);

- using a compliance server to perform compliance verification to confirm that said software file complies with a predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, “... the sequencing program and database determine if the processor, hard drive, monitor, and software contained in the system descriptor record of FIG. 3B have corresponding entries and corresponding integers specified by CompID in Component table 108.”*); and

- transferring said software file to a target information handling system after verifying that said software file complies with a predetermined set of rules (*see Figure 1: 170 and 190; Figure 2: 202; Column 10: 61-64, “... the output files reside upon the server 202 or file server 190, where they can be used to direct the execution of the software installation and testing steps upon target computer system 160.”*).

However, Amberg et al. do not disclose:

- generating commands to control the automatic downloading of software images of said software file to a target information handling system.

Feinman discloses:

- generating commands to control the automatic downloading of software images of said software file to a target information handling system (*see Figure 1B: 11; Figure 7: 100; Column 3: 3-5, "... the server 11 shown has application programs which require automatic delivery to the remote clients 13." and 44-67 through Column 4: 1-4, "Once the application program(s) have been packed up, the next step is to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point and the time that each application program is to be delivered (FIG. 3)."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include generating commands to control the automatic downloading of software images of said software file to a target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to send the application program to the remote site at a specified time, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Feinman – Column 1: 23-32*).

As per **Claim 10**, the rejection of **Claim 8** is incorporated; and Amberg et al. further disclose:

Art Unit: 2100

- generating a non-compliance notice message upon detection that said software file does not comply with said predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, "If a component is not legal (i.e. if a component in the system descriptor record is not contained in Component table 108), an error flag is raised."*).

As per **Claim 12**, the rejection of **Claim 8** is incorporated; and Amberg et al. further disclose:

- confirming proper operation of said software file on said target information handling system (*see Figure 2: 202; Figure 11; Column 12: 1-3, "... this program is called Runstep and is located on step disk 150 in the embodiment of FIG. 1 and on file server 202 in the embodiment of FIG. 2."* and 62-67 through Column 13: 1-12, "... the Runstep program fist checks to see if a file named Re_Run.bat exists. A Re_Run.bat file is created before any command is executed from a step sequence and is removed after successful completion of the command. The existence of Re_Run.bat indicates to the Runstep program in module 900 that the last command run was not successfully completed.").

However, Amberg et al. do not disclose:

- confirming the download of said software file to said target information handling system.

Feinman discloses:

- confirming the download of said software file to said target information handling system (*see Figure 5: 70; Column 3: 40-43, "The list will be used later on by the automatic installation system to ensure that all files for an installation were transferred and unpacked*

Art Unit: 2100

successfully.”; Column 5: 18-21, “If the file transfer fails in step 70, an error code is trapped and an error message is placed into two of three logs maintained by the automatic installation system as shown in step 74.”).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include confirming the download of said software file to said target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to validate that the application delivery was successful, so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Feinman – Column 1: 33-37*).

As per **Claim 13**, the rejection of **Claim 8** is incorporated; and Amberg et al. further disclose:

- notifying a manager regarding the status of said software file within a software distribution system (*see Column 14: 20-25, “... results from the software installation and testing may be logged back to either file server 190 or to file server 202. The results preferably include whether all the steps were completed successfully and what types of failures (if any) were encountered.”*).

As per **Claim 15**, Amberg et al. disclose:

- a data processor (*see Column 3: 32-37, “Target system 160 might include ... a certain brand of processor ...”*); and

Art Unit: 2100

- a data storage having a software file stored thereon (*see Column 3: 32-37, "Target system 160 might include a certain brand of hard drive ..."*), said software file being transferred to said data storage by an automated software dissemination system comprising:

- a distribution server operable to receive a software file (*see Figure 1: 140; Column 3: 48-55, "... sequencing program 204 residing on step maker 140, first reads a plurality of component descriptors from descriptor file 96."* and 59-66, "... the component descriptors are included in a descriptor file called a system descriptor record which is a computer readable file containing a listing of the components, hardware and/or software components, to be installed onto target system 160.");

- a repack and script regeneration server operably connected to said distribution server, said repack and script regeneration server operable to disassemble said software file and repackage said software file with scripts for automatically controlling the transfer of said software file (*see Figure 1: 140; Column 4: 8-14, "Having retrieved the software installation and/or testing steps appropriate for target system 160, sequencing program 204 sequences the steps in a predetermined order according to sequence numbers corresponding to each step. Having sequenced the steps required for target system 160, sequencing program 204 writes a series of output files to step disk 150."*; Column 9: 56-67 through Column 10: 1-2, "The result of the joinder of the Sys_Comp table 112 and the Comp_Step table 114 is then joined with the Sys_Step_Seq table 106 which contains all the steps for family X." and "... a three-table join of Sys_Comp table 112, Comp_Step table 114, and Sys_Step_Seq table 106 yields the appropriate software installation and testing steps as well as sequencing information in the form of sequence and phase numbers to install and/or test software upon target computer system 160.");

- a compliance server operably connected to said distribution server, said compliance server being operable to perform compliance verification to confirm that said software file complies with a predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, "... the sequencing program and database determine if the processor, hard drive, monitor, and software contained in the system descriptor record of FIG. 3B have corresponding entries and corresponding integers specified by CompID in Component table 108."*); and

- a download server operable to transfer said software file to said information handling system after verifying that said software file complies with a predetermined set of rules (*see Figure 1: 170 and 190; Figure 2: 202; Column 10: 61-64, "... the output files reside upon the server 202 or file server 190, where they can be used to direct the execution of the software installation and testing steps upon target computer system 160."*).

However, Amberg et al. do not disclose:

- a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to said information handling system.

Feinman discloses:

- a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to said information handling system (*see Figure 1B: 11; Figure 7: 100; Column 3: 3-5, "... the server 11 shown has application programs which require automatic delivery to the remote clients 13."*

and 44-67 through Column 4: 1-4, "Once the application program(s) have been packed up, the next step is to identify the remote client's delivery points, the application programs that are to be delivered to each delivery point and the time that each application program is to be delivered (FIG. 3).").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include a script validation server operably coupled to said repack and script regeneration server and said distribution server, said script validation server operable to generate commands to automatically control the downloading of software images of said software file to said information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to send the application program to the remote site at a specified time, since it is not practical for someone to send the application to worldwide sites manually because many sites require late night delivery and also due to the sheer volume of user sites (*see Feinman – Column 1: 23-32*).

As per **Claim 17**, the rejection of **Claim 15** is incorporated; and Amberg et al. further disclose:

- wherein said compliance server is operable to automatically generate a non-compliance notice message upon detection that said software file does not comply with said predetermined set of rules (*see Figure 1: 100; Column 9: 9-16, "If a component is not legal (i.e. if a component in the system descriptor record is not contained in Component table 108), an error flag is raised."*).

As per **Claim 19**, the rejection of **Claim 15** is incorporated; and Amberg et al. further disclose:

- a test control server operable to verify proper operation of said software file on said target information handling system (*see Figure 2: 202; Figure 11; Column 12: 1-3, "... this program is called Runstep and is located on step disk 150 in the embodiment of FIG. 1 and on file server 202 in the embodiment of FIG. 2." and 62-67 through Column 13: 1-12, "... the Runstep program fist checks to see if a file named Re_Run.bat exists. A Re_Run.bat file is created before any command is executed from a step sequence and is removed after successful completion of the command. The existence of Re_Run.bat indicates to the Runstep program in module 900 that the last command run was not successfully completed."*).

However, Amberg et al. do not disclose:

- a test control server operable to confirm the download of said software file to said target information handling system.

Feinman discloses:

- a test control server operable to confirm the download of said software file to said target information handling system (*see Figure 5: 70; Column 3: 40-43, "The list will be used later on by the automatic installation system to ensure that all files for an installation were transferred and unpacked successfully."; Column 5: 18-21, "If the file transfer fails in step 70, an error code is trapped and an error message is placed into two of three logs maintained by the automatic installation system as shown in step 74."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Feinman into the teaching of Amberg et al. to include a test control server operable to confirm the download of said software file to said target information handling system. The modification would be obvious because one of ordinary skill in the art would be motivated to validate that the application delivery was successful, so that any corrective action could be taken in the event that there is a problem, which is especially important if the installation system is to run unattended (*see Feinman – Column 1: 33-37*).

As per **Claim 20**, the rejection of **Claim 15** is incorporated; and Amberg et al. further disclose:

- wherein said distribution server is operable to notify a manager regarding the status of said software file within a software distribution system (*see Column 14: 20-25, “... results from the software installation and testing may be logged back to either file server 190 or to file server 202. The results preferably include whether all the steps were completed successfully and what types of failures (if any) were encountered.”*).

2. **Claims 4, 11, and 18** are rejected under 35 U.S.C. 103(a) as being unpatentable over Amberg et al. (US 5,991,543) in view of Feinman (US 6,075,943) as applied to Claims 1, 8, and 15 above, and further in view of Tso et al. (US 6,088,803).

As per **Claim 4**, the rejection of **Claim 1** is incorporated; however, Amberg et al. and Feinman do not disclose:

- wherein said distribution server is operable to scan said software file for viruses.

Tso et al. disclose:

- wherein said distribution server is operable to scan said software file for viruses (*see Figure 2: 40; Column 2: 62-67 through Column 3: 1-5, "... network device 4 invokes virus checker 5, which in turn performs its preconfigured virus scan processing with the requested file as input (Step 40)."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Tso et al. into the teaching of Amberg et al. to include wherein said distribution server is operable to scan said software file for viruses. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize breaches in system integrity (*see Tso et al. – Column 1: 27-28*).

As per **Claim 11**, the rejection of **Claim 8** is incorporated; however, Amberg et al. and Feinman do not disclose:

- scanning said software file for viruses.

Tso et al. disclose:

- scanning said software file for viruses (*see Figure 2: 40; Column 2: 62-67 through Column 3: 1-5, "... network device 4 invokes virus checker 5, which in turn performs its preconfigured virus scan processing with the requested file as input (Step 40)."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Tso et al. into the teaching of Amberg et al. to include scanning said software file for viruses. The modification would be obvious because one

Art Unit: 2100

of ordinary skill in the art would be motivated to minimize breaches in system integrity (*see Tso et al. – Column 1: 27-28*).

As per **Claim 18**, the rejection of **Claim 15** is incorporated; however, Amberg et al. and Feinman do not disclose:

- wherein said distribution server is operable to scan said software file for viruses.

Tso et al. disclose:

- wherein said distribution server is operable to scan said software file for viruses (*see Figure 2: 40; Column 2: 62-67 through Column 3: 1-5, “... network device 4 invokes virus checker 5, which in turn performs its preconfigured virus scan processing with the requested file as input (Step 40).”*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Tso et al. into the teaching of Amberg et al. to include wherein said distribution server is operable to scan said software file for viruses. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize breaches in system integrity (*see Tso et al. – Column 1: 27-28*).

3. **Claims 7 and 14** are rejected under 35 U.S.C. 103(a) as being unpatentable over Amberg et al. (US 5,991,543) in view of Feinman (US 6,075,943) as applied to Claims 1 and 8 above, and further in view of Karasudani et al. (US 6,378,054).

As per **Claim 7**, the rejection of **Claim 1** is incorporated; however, Amberg et al. and Feinman do not disclose:

- an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon.

Karasudani et al. disclose:

- an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon (*see Column 11: 60-61, "When the mth data file is added to the archive database ..."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Karasudani et al. into the teaching of Amberg et al. to include an archive server, wherein said repack and script regeneration server is operable to transfer copies of said software file to said archive server for storage thereon. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize damages by immediately recovering a data file in the event of problems, such as the loss of data (*see Karasudani et al. – Column 1: 26-29*).

As per **Claim 14**, the rejection of **Claim 8** is incorporated; however, Amberg et al. and Feinman do not disclose:

- transferring copies of said software file to an archive server for storage thereon.

Karasudani et al. disclose:

- transferring copies of said software file to an archive server for storage thereon (*see Column 11: 60-61, "When the mth data file is added to the archive database ..."*).

Art Unit: 2100

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Karasudani et al. into the teaching of Amberg et al. to include transferring copies of said software file to an archive server for storage thereon. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize damages by immediately recovering a data file in the event of problems, such as the loss of data (*see Karasudani et al. – Column 1: 26-29*).

(10) Response to Argument

In the Appeal Brief, Appellant argues:

a) Examiner originally rejected dependent claims 2, 9 and 16 under 35 U.S.C. § 103(a) based on the combination of Amberg as modified by Feinman. Specifically, Examiner alleged that the features recited in originally submitted dependent claims 2, 9, and 16 are disclosed by Amberg in Col. 9, lines 9-16. The cited portion of the Amberg reference describes a system wherein various components are compared to a "component table" prior to installation on an information handling system. The cited portion of Amberg does not provide a teaching of a compliance server that is operable to perform compliance verification to confirm that a software file complies with a predetermined set of software rules, as recited in amended independent claims 1, 8 and 15.

(See Appeal Brief – page 5)

Examiner's response:

Art Unit: 2100

a) Examiner disagrees. Appellant's arguments are not persuasive for at least the following reasons:

First, Appellant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

Second, Amberg clearly discloses "a compliance server operably connected to said distribution server, said compliance server being operable to perform compliance verification to confirm that said software file complies with a predetermined set of software rules" (*see Figure 1: 100; Column 9: 9-16, "... the sequencing program and database determine if the processor, hard drive, monitor, and software contained in the system descriptor record of FIG. 3B have corresponding entries and corresponding integers specified by CompID in Component table 108."*). Thus, the database server (compliance server) contains the Component table (predetermined set of software rules). The database determines if the components (*e.g.*, processor, hard drive, monitor, software, etc.) contained in the system descriptor (software file) have corresponding entries in the Component table.

Third, the claims recite only a "predetermined set of software rules" with no further clarification on the claim scope of the limitation "software rules" as intended by the Appellant to cover. The broadly recited limitation can be interpreted in many different ways, for example, rules implemented in software or rules about the software. Thus, as the claims are interpreted as broadly as their terms reasonably allow (*see MPEP § 2111.01 I*), the interpretation of a broad limitation of "software rules" as attributes of the Component table and the like by one of ordinary skill in the art is considered to be reasonable by its plain meaning.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the Examiner in the Related Appeals and Interferences section of this Examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Qing Chen

/Q. C./

Examiner, Art Unit 2191

Conferees:

Wei Zhen

/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191

Eddie Lee

/Eddie Lee/

Supervisory Patent Examiner, TC 2100